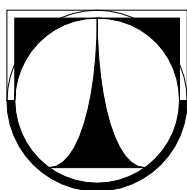


**TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky, informatiky a mezioborových studií



**BAKALÁŘSKÁ PRÁCE**

Liberec 2012

**Petr Hliza**

**TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2646 – Informační technologie

Studijní obor: 1802R007 – Informační technologie

**Metoda konečných objemů s prostorově  
adaptivním časovým krokem**

**Finite Volume Method with Spatially  
Variable Time Step**

**Bakalářská práce**

Autor: **Petr Hlída**

Vedoucí práce: Mgr. Jan Březina, Ph.D.

**V Liberci 17. 5. 2012**



## **Prohlášení**

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Dne 17. 5. 2012

Petr Hlída

## **Poděkování**

Rád bych touto cestou poděkoval panu Mgr. Janu Březinovi za jeho veškerý čas, který mi věnoval, za zasvěcení do problematiky, kterou se sám vědecky zabývá, a za užitečné rady, které mi pomohly tuto práci sepsat.

## Abstrakt

Tato bakalářská práce pojednává o metodě konečných objemů, která se využívá při výpočtu koncentrací látky proudící v podzemním puklinovém prostředí nasyceném vodou. Důležitým faktorem je časový krok, který musí pro celou transportní matici splňovat tzv. CFL podmínku. Ta v případě heterogenního pole mikroskopických rychlostí způsobí malý časový krok odpovídající nejvyšší rychlosti v oblasti. Použití stejného časového kroku pro celou oblast je proto neefektivní a vede i k větší numerické difúzi v oblastech s nižšími rychlostmi. Z tohoto důvodu jsme navrhli novou metodu, která počítá v oblastech s nižšími rychlostmi s delším časovým krokem než v oblasti s vyššími rychlostmi. Pro ověření správnosti naší metody jsme ji otestovali pro úlohu transportu látky jednorozměrnou oblastí s heterogenní porozitou.

Klíčová slova: Metoda konečných objemů, transport látky, CFL podmínka, časový krok, numerická difúze.

## Abstract

This bachelor work discuss finite volume method, which is used for calculating concentrations of substance flowing in an underground water-saturated environment with fissures. An important factor is the time step, which has to satisfy the CFL condition for the whole transport matrix. In the case of heterogeneous microscopic velocity field this leads to the small time step corresponding to the highest velocity in the area. Using the same time step for the whole area is ineffective and causes greater numerical diffusion in regions with lower velocities. Therefore we designed a new method that counts with longer time step in regions with lower velocities and in regions with high velocities it uses shorter time step. We have tested our method on one-dimensional area with heterogeneous porosity.

Keywords: finite volume method, substance transport, CFL condition, time step, numerical diffusion.

# Obsah

Prohlášení.....	3
Poděkování.....	4
Abstrakt.....	5
Obsah .....	6
1 Úvod .....	7
2 Popis úlohy transportu .....	8
3 Diskretizace metodou konečných objemů .....	10
3.1 CFL podmínka .....	13
4 Nová metoda pro rozdělenou transportní matici .....	14
4.1 Metoda multiplikativní aproximace .....	14
5 Vlastní program .....	17
5.1 Celkový přehled funkcí programu .....	17
5.1.1 Zápis generovaných dat do souboru .....	17
5.2 Graf .....	18
5.3 Určení toků a objemů elementů .....	20
5.4 Řídká transportní matice .....	20
5.4.1 Rozdělení transportní matice .....	21
5.4.2 Tvorba permutačního vektoru a dělení matice A.....	21
5.4.3 Násobení řídké matice vektorem .....	26
6 Testování implementovaných metod.....	28
6.1 Testovací úloha .....	28
6.2 Test doby oběhu pro $\alpha = 0$ .....	29
6.3 Test doby oběhu pro $\alpha \geq 0$ .....	30
6.4 Test bilance hmoty .....	31
6.5 Porovnání implementovaných metod.....	33
6.6 Porovnání výpočetních časů.....	34
7 Závěr.....	37
Seznam použité literatury .....	38

# 1 Úvod

Hlavním cílem této práce bylo seznámit se s problematikou transportu látky proměnným rychlostním polem. Jedná se například o transport látky v podzemním puklinovém prostředí nasyceném vodou, kde je v různých místech horniny odlišná porozita. Čím je v daném místě porozita nižší, tím větší mikroskopickou rychlostí tam voda proudí a naopak.

Pro výpočet koncentrací transportované látky využijeme metodu konečných objemů (kapitola 3). Sledovanou oblast si pro zjednodušení rozdělíme na síť elementů, kde každý element má svou vlastní porozitu, objem a rychlost toku vody. Na základě této sítě vytvoříme transportní matici, kterou využijeme při výpočtech koncentrací látky v konkrétních elementech oblasti. Výpočet se provádí pomocí časového kroku, který musí splňovat tzv. CFL podmínku. Jelikož je časový krok malý a homogenní na celé oblasti, výpočet koncentrací je časově náročný a způsobuje větší numerickou difúzi v oblastech s nižšími mikroskopickými rychlostmi.

Další částí práce bylo navržení nové metody pro výpočet koncentrací, která by částečně odstranila nedostatky původní metody konečných objemů. Nová metoda využívá transportní matici rozdělenou na dvě menší matice na základě rychlostí toků na jednotlivých elementech. Při výpočtu koncentrací pro větší matici, obsahující nízké rychlosti, počítá s delším časovým krokem, na rozdíl od menší matice s vysokými rychlostmi, pro kterou používá kratší časový krok.

V páté kapitole se zabývám vytvořením speciálního programu v jazyce Java, který slouží k otestování obou metod konečných objemů. Program vytváří graf pro jednorozměrnou oblast, kterou rozdělí na zadaný počet elementů. Pro všechny elementy spočítá hodnoty objemů, porozity a mikroskopických rychlostí. Z těchto dat potom vytvoří transportní matici a provede výpočet koncentrací metodou konečných objemů. Program také umí rozdělit transportní matici na dvě menší matice, potřebné pro výpočet koncentrací naší novou metodou. Generované vektory koncentrací jsou ukládány do souborů s příponou .vtk, které můžeme zobrazit ve vizualizačním nástroji Paraview.

V kapitole 6 je uveden přehled všech testů, které jsme provedli pro porovnání naší nové metody s původní metodou konečných objemů. Testy zároveň sloužily k ověření, zda nová metoda funguje správně, zda výpočet koncentrací touto metodou trvá kratší dobu a jestli dochází k menší numerické difúzi.



## 2 Popis úlohy transportu

Úloha se zabývá transportem látky v podzemní vodě v horninovém prostředí. V praxi je však velmi složité zjistit všechny geostrukturální, geofyzikální a geochemické vlastnosti v zkoumané oblasti (značíme  $\Omega$ ). Pomocí jadrových vrtů, laboratorních, terénních a dalších měření získáme potřebná data jen pro několik málo míst z dané oblasti. Pro celý zbytek oblasti  $\Omega$  pak musíme předpokládat, že v něm jsou zkoumané vlastnosti podobné těm naměřeným.

Důležitou vlastností horninového materiálu v oblasti  $\Omega$  je jeho porozita. *Porézním prostředím (horninovým materiálem) nazýváme strukturu složenou ze zrn nebo vláken pevné látky (matrice), mezi nimiž je volný prostor (póry), který může být zaplněn plynem (vzduchem) nebo kapalinou, podle toho rozlišujeme saturované nebo nesaturované prostředí. Přesný popis dějů v porézním prostředí je z důvodu mikroskopické struktury velmi složitý, proto se pracuje s aproximací porézního prostředí jako spojitého prostředí (kontinua). Všechny veličiny jsou pak uvažovány vždy jako střední hodnoty (průměr) přes určitý elementární objem.* [2]

Pro spojitou reprezentaci porézního prostředí si definujeme konečný objem  $E_i$ . Tento objem musí být řádově větší než je rozměr zrn pevné látky, ale zároveň dostatečně malý vzhledem k velikosti zkoumané oblasti. Celou oblast si tak rozdělíme na síť jednotlivých konečných objemů s různou porozitou. Porozita konečného objemu se rovná poměru objemu pórů vzhledem k celému objemu materiálu tohoto konečného objemu.

Pro každý konečný objem také musí platit zákon o zachování hmoty – množství látky, které do daného konečného objemu přiteče za jednotku času, se rovná množství látky, které z něj za stejnou dobu vyteče ven. Z tohoto tvrzení vyplývá, že rychlost šíření látky vtékající do konečného objemu se musí rovnat rychlosti odtoku látky. Pokud by rychlost nebyla stejná na všech konečných objemech  $E_i$  pozorované oblasti  $\Omega$ , muselo by někde docházet ke ztrátě určitého množství látky, nebo by naopak látka někde přibývala. My však předpokládáme, že zkoumaná oblast  $\Omega$  je uzavřená, takže rychlostní pole  $\mathbf{v}$  bude konzervativní. Platí tedy:

$$\operatorname{div} \mathbf{v} = 0 \quad \text{na } \Omega \quad (2.1)$$

Tato rychlost představuje makroskopickou rychlost toku látky na oblasti  $\Omega$ , skutečná mikroskopická rychlost vody v pórech závisí na porozitě. Čím menší je porozita v konečném objemu  $E_i$ , tím rychleji v něm bude voda proudit. Naopak v místech s vysokou porozitou je propustnost vody velká, takže skutečná rychlost proudění bude nižší.

Další sledovanou hodnotou je koncentrace látky  $c$  na jednotlivých elementech sítě (konečné objemy). Koncentrace se mění v čase s tím, jak se látka šíří sledovanou oblastí z jednoho konečného objemu do druhého. Změna koncentrací na jednotlivých elementech je závislá na rychlosti proudění látky a na objemu a porozitě jednotlivých konečných objemů.

### 3 Diskretizace metodou konečných objemů

Uvažujeme tuto spojitou rovnici pro transport látky (tzv. rovnici kontinuity [1])

$$\partial_t(c\gamma) + \operatorname{div}(c\mathbf{v}) = 0 \quad \text{na } \Omega, \quad (3.1)$$

kde  $c$  je koncentrace rozpuštěné látky v oblasti  $\Omega$ ,  $\gamma$  je porozita prostředí a  $\mathbf{v}$  je rychlostní pole. Zkoumanou oblast  $\Omega$  si rozdělíme na síť  $n$  trojúhelníků, kde každý trojúhelník reprezentuje jeden konečný objem  $E_i$ . Uvažujeme pole koncentrací i porozity konstantní na každém elementu, tj.  $c = c_i$  a  $\gamma = \gamma_i$  na  $E_i$ . Integrací per partes přes element  $E_i$  dostaneme

$$\int_{E_i} \partial_t(c_i\gamma_i) + \int_{\partial E_i} c\mathbf{v}\mathbf{n} = 0, \quad (3.2)$$

kde  $\partial E_i$  představuje hranici  $E_i$ , v našem případě trojúhelníku. Druhý člen rovnice můžeme zapsat jako

$$\int_{\partial E_i} c\mathbf{v}\mathbf{n} = \sum_{e \in \partial E_i} c_e \int_e \mathbf{v}\mathbf{n}. \quad (3.3)$$

Integrál  $\int_e \mathbf{v}\mathbf{n}$  pak představuje tok vody přes hranu  $e$ , s normálovým vektorem  $\mathbf{n}$  směřujícím ven z  $E_i$ . Koncentrace  $c_e$  na hranách bereme z elementu, ze kterého vytéká voda na hraně  $e$ , čemuž říkáme upwind. Tento tok z elementu  $E_i$  do elementu  $E_j$  značíme  $q_{ij}$  a vypočteme ho takto:

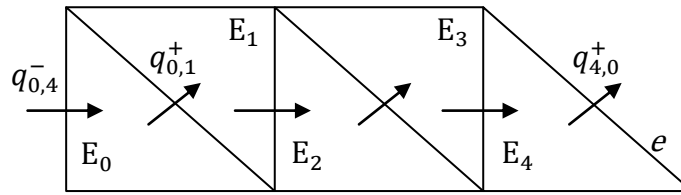
$$q_{ij} = \int_{e_{ij}} \mathbf{v}\mathbf{n}. \quad (3.4)$$

Pokud je tok  $q_{ij} > 0$ , pak bereme koncentraci na hraně  $e$  z elementu  $E_i$ , tj.  $c_e = c_i$ .

Další diskretizací rovnice (3.2) explicitní Eulerovou metodou v čase a metodou konečných objemů s upwindem získáme rovnici

$$\frac{|E_i|\gamma_i}{\tau}(c_i^{t+\tau} - c_i^t) + \sum_{j=0}^n q_{ji}^- c_i^t - \sum_{j=0}^n q_{ji}^+ c_j^t = 0, \quad (3.5)$$

kde  $|E_i|$  je objem konečného objemu  $E_i$  a  $\gamma_i$  je jeho porozita,  $\tau$  je časový krok,  $c_i^t$  značí koncentraci látky na konečném objemu  $E_i$  v čase  $t$ . Dále označíme  $q_{ij}^+ = \max(q_{ij}, 0)$  jako kladou část toku  $q_{ij}$  a  $q_{ij}^- = -\min(q_{ij}, 0)$  jako zápornou část. Jinými slovy to znamená, že pokud tok směřuje z konečného objemu  $E_i$  do  $E_j$ , pak  $q_{ij}^+$  udává, jaké množství látky přeteče z  $E_i$  do  $E_j$  za jednotku času. Platí tedy  $q_{ij}^+ = q_{ji}^- > 0$ . Naopak pro opačné toky pak platí  $q_{ij}^- = q_{ji}^+ = 0$  (z  $E_j$  do  $E_i$  nic neteče).



**Obr. 3.1: Znázornění toků na pěti elementech**

Pro výpočet koncentrací v novém čase  $t + \tau$  můžeme napsat rovnici

$$c^{t+\tau} = (I + \tau A)c^t, \quad (3.6)$$

kde  $I$  je jednotková matice,  $\tau$  časový krok a  $A$  je transportní matice pro oblast  $\Omega$ . Rovnici (3.6) pro výpočet koncentrací nazveme jako **FVM** (od anglického spojení Finite Volume Method). Jednotlivé prvky transportní matice  $A$  se vypočítají takto:

$$A_{ij} = \begin{cases} \frac{q_{ji}^+}{|E_i|\gamma_i} & \text{pro } i \neq j \\ -\sum_{k=1}^n \frac{q_{ki}^-}{|E_i|\gamma_i} = -\sum_{k=1}^n \frac{|E_k|\gamma_k}{|E_i|\gamma_i} A_{ki} & \text{pro } i = j \end{cases} \quad (3.7)$$

Ve vztahu (3.7) značí  $i$  řádek matice a  $j$  sloupec. Na diagonále matice  $A$  tak bude na každém řádku  $i$  uložena záporná suma všech toků, které tečou pryč z  $E_i$ . Ve sloupci  $j$  řádku  $i$  pak bude uložena hodnota toku  $q_{ji}^+$ , která udává velikost toku přitékajícího z  $E_j$ . Jelikož je naše síť složena z trojúhelníků, maximální počet toků pro jeden element sítě je roven třem. Pokud budou do  $E_i$  přitékat dva toky, budou na řádku  $i$  matice  $A$  tři nenulové prvky (jeden na diagonále, dva mimo diagonálu). Pokud by byl přítok do  $E_i$  jen jeden, v matici  $A$  budou na řádku  $i$  pouze dva nenulové prvky. Zároveň musí pro každý řádek platit již zmíněný zákon o zachování hmoty, takže bilance toků na řádku musí být vždy nulová (proto mají sumy odtoků na diagonále záporná znaménka).

Zde je příklad jednoduché transportní matice pro síť skládající se z pěti elementů, kde pro každý element platí, že jeho objem  $|E_i| = 1$  a jeho porozita  $\gamma_i = 1$ . Makroskopická rychlost šíření látky v oblasti je  $v = 5$ .

		$j \rightarrow$				
		0	1	2	3	4
$i$ ↓	0	-5	0	0	0	5
	1	5	-5	0	0	0
	2	0	5	-5	0	0
	3	0	0	5	-5	0
	4	0	0	0	5	-5

**Obr. 3.2: Transportní matice pro pět elementů**

Jelikož jsou objemy i porozita rovny jedné a jsou konstantní pro všechny elementy sítě, je skutečná velikost toku na elementech rovna makroskopické rychlosti  $v$ . Z této transportní matice  $A$  můžeme například zjistit, že do konečného objemu  $E_0$  přitéká látka rychlostí 5 z elementu  $E_4$  ( $A_{0,4} = 5$ ) a zároveň stejné množství látky z  $E_0$  odtéká pryč ( $A_{0,0} = -5$ ). Platí tedy bilance toků (rychlostí) na elementu  $E_0$ . Tok látky v tomto případě směřuje z elementu  $E_0$  do  $E_4$ , odkud teče zpět do  $E_0$ . Tato síť konečných objemů tak utváří uzavřenou oblast  $\Omega$ .

Transportní matice se používá pro výpočet změn koncentrací na jednotlivých elementech sledované oblasti podle vztahu (3.6). Abychom při výpočtu koncentrací nezískali záporné hodnoty koncentrací látky, je nutné dodržet takzvanou CFL podmínku.

### 3.1 CFL podmínka

CFL podmínka je nutnou podmínkou stability, která při numerickém řešení parciálních diferenciálních rovnic (transport látky) zaručuje kladné hodnoty koncentrací. Tato podmínka určuje, jak moc krátký musí být časový krok  $\tau$ , abychom získali správný výsledek. V našem případě musí pro každý diagonální prvek matice  $A$  platit nerovnice

$$1 + \tau A_{ii} \geq 0. \quad (3.8)$$

Aby byla tato podmínka splněna, vypočteme časový krok  $\tau$ .

$$\tau = \frac{1}{\max |A_{ii}|} \quad (3.9)$$

Pro matici na obrázku 3.2 je  $\tau = 0,2$ . Za ideálních podmínek, když je skutečná rychlost na elementech shodná s makroskopickou rychlostí, se daná koncentrace přesune z jednoho konečného objemu do sousedního právě za tento časový krok. Pokud by v čase  $t = 0$  byl vektor koncentrací  $c^t = [1,0,0,0,0]$ , pak za jeden časový krok přeteče celé množství látky z  $E_0$  do  $E_1$ . V čase  $t = 0,2$  by tak  $c^{t+\tau} = [0,1,0,0,0]$ . Z toho plyne, že za dobu  $t = 1$  by koncentrace látky dorazila zpět do  $E_0$ .

V reálných příkladech však objemy, porozita a rychlosti na jednotlivých elementech konstantní nejsou. Čím vyšší rychlosti se budou na diagonále transportní matice  $A$  vyskytovat, tím kratší bude výsledný časový krok. I kdyby byly všechny rychlosti relativně malé a jen jedna z nich řádově vyšší, časový krok se řádově zmenší, což ovlivní výpočetní čas pro určení koncentrací pro celou matici  $A$  (ne pouze pro řádek s vysokou rychlostí). Vhodnější by bylo, kdybychom mohli pro část matice s nízkými rychlostmi používat delší časový krok a pro zbytek matice s vysokými rychlostmi použít kratší časový krok. Tato úvaha vede k rozdělení transportní matice na dvě části a následnému odvození nové metody pro výpočet koncentrací.

## 4 Nová metoda pro rozdělenou transportní matici

Předpokládáme, že bychom transportní matici  $A$  rozdělili na dvě menší matice  $A_1$  a  $A_2$  tak, aby platilo  $A = A_1 + A_2$ . V matici  $A_1$  bychom chtěli mít jen hodnoty odpovídající vysokým rychlostem, v  $A_2$  pak všechny ostatní. Kvůli zachování hmoty je nejjednodušší rozdělit matici po celých sloupcích. Zařazení provedeme podle maximálního prvku na diagonále matice  $A$ .

$$\begin{aligned} E_i &\in A_1 && \text{pro } |A_{ii}| > \frac{1}{2} \max_j |A_{jj}| \\ E_i &\in A_2 && \text{pro } |A_{ii}| \leq \frac{1}{2} \max_j |A_{jj}| \end{aligned} \quad (4.1)$$

### 4.1 Metoda multiplikativní aproximace

Jelikož v této metodě budeme počítat s rozdělenou transportní maticí, musíme příslušně upravit rovnici pro výpočet koncentrace. S maticí  $A_1$  budeme kvůli dvakrát větším rychlostem počítat dvakrát tak často než s maticí  $A_2$ . Jeden krok této metody se proto bude rovnat dvěma časovým krokům na matici  $A_1$  a jednomu dvojkroku na matici  $A_2$ . Ze vzorečku pro výpočet koncentrací metodou 3.6 získáme matici  $T = (I + \tau A)$ . Pro dva časové kroky pak s použitím rozdělených matic získáme následující rovnici.

$$T^2 = (I + \tau A)^2 = I + 2\tau(A_1 + A_2) + \tau^2(A_1^2 + A_1A_2 + A_2A_1 + A_2^2) \quad (4.2)$$

Výše uvedenou rovnici vedoucí práce aproximoval rovnici

$$T^2 \approx T_m = (I + 2\tau A_2 + \tau A_1)(I + \tau A_1). \quad (4.3)$$

Její chyba oproti rovnici (4.2) se rovná

$$T^2 - T_m = \tau^2(A_2^2 + A_1A_2 - A_2A_1). \quad (4.4)$$

Na základě rovnice (4.3) spočítáme koncentrace v čase  $t + 2\tau$  ve dvou krocích.

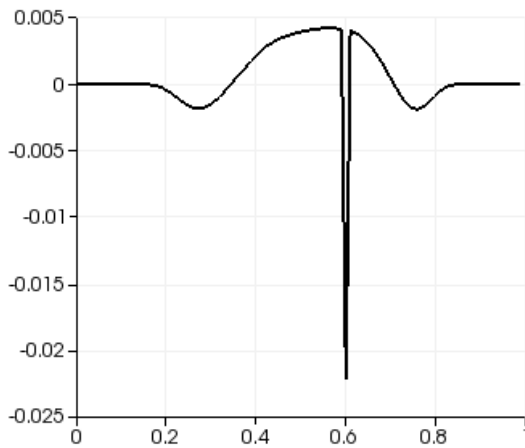
$$1) \quad c^{t+\tau} = c^t + \tau A_1 c^t \quad (4.5)$$

$$2) \quad c^{t+2\tau} = c^{t+\tau} + 2\tau A_2 c^{t+\tau} + \tau A_1 c^{t+\tau} \quad (4.6)$$

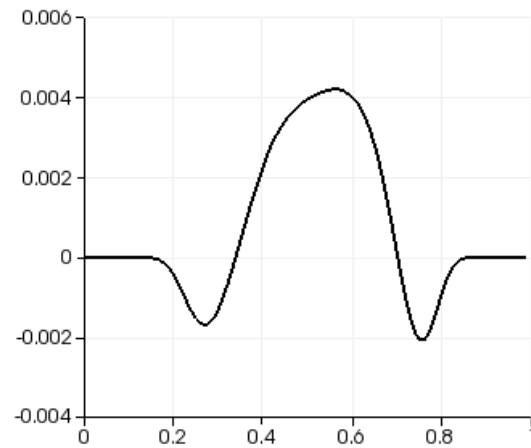
V prvním kroku výpočtu násobíme vektor koncentrací pouze maticí  $A_1$ , která bude v ideálním případě malá, takže výpočet bude trvat mnohem kratší dobu, než kdybychom násobili celou maticí  $A$ . Ve druhém kroku násobíme oběma částmi matice  $A$ , takže tento krok by měl trvat přibližně stejně dlouho jako standardní krok pro výpočet koncentrace s celou transportní maticí  $A$ .

Od této metody tedy očekáváme až dvojnásobné zrychlení doby výpočtu oproti původní metodě za předpokladu, že matice  $A_1$  bude několikrát menší než matice  $A$ . Navíc by metoda měla poskytovat výsledky s menší numerickou difúzí. Ověření těchto hypotéz testujeme v kapitole 6.

Při testování nové aproximované metody (AM) s použitím rovnice (4.3) jsme ovšem našli zásadní chybu, která způsobovala, že v některých místech vypočteného vektoru koncentrací se vyskytovaly výrazně nižší nebo vyšší hodnoty oproti okolním hodnotám. Tato chyba je dobře vidět v grafu na obrázku 4.1, kde je zobrazena křivka rozdílu koncentrací vypočtených novou AM metodou a původní metodou FVM.



**Obr. 4.1: Rozdíl koncentrací AM – FVM**



**Obr. 4.2: Rozdíl koncentrací AFVM - FVM**



Po podrobné analýze výpočtu jsme zjistili, že tyto viditelné „zuby“ v grafu jsou přesně v místech, kde se překrývají matice  $A_1$  a  $A_2$ . Provedením dalších výpočtů a zkoušek jsme dospěli k zjištění, že je potřeba do výpočtu metody AFVM začlenit člen  $A_1A_2$  a odstranit člen  $2\tau A_2A_1$ . Tím jsme došli k této rovnici:

$$T_{m_2} = I + 2\tau(A_2 + A_1) + \tau A_1(\tau A_1 + \tau A_2) \quad (4.7)$$

s chybou:

$$T^2 - T_{m_2} = \tau^2(A_2^2 + A_2A_1) \quad (4.8)$$

Tuto upravenou metodu (4.7) označíme jako **AFVM** (Approximate Finite Volume Method) a příslušné koncentrace v čase  $t + 2\tau$  vypočteme v následujících dvou krocích.

$$1) \quad \tilde{c} = (\tau A_1 + \tau A_2)c^t \quad (4.9)$$

$$2) \quad c^{t+2\tau} = c^t + 2\tilde{c} + \tau A_1\tilde{c} \quad (4.10)$$

Jak je vidět v grafu na obrázku 4.2, křivka rozdílu koncentrací, vypočtených metodou AFVM a metodou FVM, má hladký průběh, takže se nám nalezené defekty podařilo úspěšně odstranit.

## 5 Vlastní program

V této části se budu věnovat vytvoření testovacího programu v jazyce Java a implementaci některých důležitých prvků potřebných pro aplikaci obou dvou metod konečných objemů.

### 5.1 Celkový přehled funkcí programu

Program slouží pro výpočet koncentrace látky v oblasti  $\Omega$  v čase  $t$  pomocí implementovaných metod konečných objemů – FVM a AFVM. Program umí na základě zadaných dat vytvořit graf pro síť elementů reprezentujících oblast  $\Omega$ , včetně výpočtu objemů, porozity a velikosti toků na jednotlivých elementech. Z těchto dat potom dokáže sestavit transportní matici  $A$  a spočítat změny koncentrací na elementech v čase  $t$  metodou FVM. Dále program dokáže rozdělit transportní matici  $A$  na menší matice  $A_1$  a  $A_2$ , které pak využije při výpočtu koncentrací metodou AFVM.

#### 5.1.1 Zápis generovaných dat do souboru

Důležitou funkcí programu je možnost zápisu vypočtených dat do souborů s příponou .vtk, které je možné otevřít ve vizualizačním nástroji Paraview. V něm můžeme zobrazit tok látky oblastí  $\Omega$  jako jednoduchou animaci, provést potřebné kontroly a nakonec porovnat obě dvě zkoumané metody konečných objemů.

Na obrázku 5.1 je ukázka obsahu souboru, který byl vygenerován pro oblast  $\Omega$  rozdělenou na 10 elementů na jednom řádku. Na řádcích 1 – 4 nalezneme hlavičku souboru, která určuje i typ zobrazené sítě. Řádek 5 udává rozměry sítě, první číslo značí počet elementů na řádku (osa  $x$ ), druhé číslo počet řádků elementů (osa  $y$ ), třetí číslo je pro osu  $z$  a je vždy rovno jedné, jelikož nezobrazujeme trojrozměrné oblasti. Pro tyto tři hodnoty ( $x, y, z$ ) jsou dále na řádcích 6 – 11 zapsány souřadnice jednotlivých elementů na příslušných osách. Na řádku 12 je uložen počet hodnot (shodný s počtem elementů), které chceme zobrazit. Řádek 13 udává, kolik datových polí ukládáme. Jejich názvy, počet hodnot, datový typ a konkrétní hodnoty jsou uloženy na řádcích 14 – 19.

```

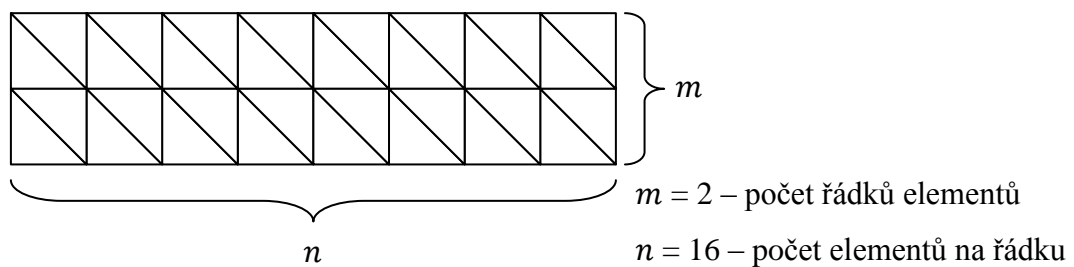
1.  # vtk DataFile Version 3.0
2.  vtk output
3.  ASCII
4.  DATASET RECTILINEAR_GRID
5.  DIMENSIONS 10 1 1
6.  X_COORDINATES 10 float
7.  0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
8.  Y_COORDINATES 1 float
9.  0.25
10. Z_COORDINATES 1 float
11. 0
12. POINT_DATA 10
13. FIELD FieldData 3
14. conc_OTS 1 10 float
15. 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
16. volume_OTS 1 10 float
17. 0.025 0.016493848884661183 ...
18. mass_OTS 1 10 float
19. 0.025 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

**Obr. 5.1: Struktura .vtk souboru**

## 5.2 Graf

Prvním krokem při tvorbě programu bylo vytvoření grafu, který by reprezentoval oblast  $\Omega$ . Oblast  $\Omega$  o nějaké šířce  $x$  a výšce  $y$  rozdělíme na  $m \times n$  elementů, kde  $m$  udává počet řádků elementů,  $n$  pak počet elementů na řádku. Výsledná oblast pak vypadá přibližně takto.



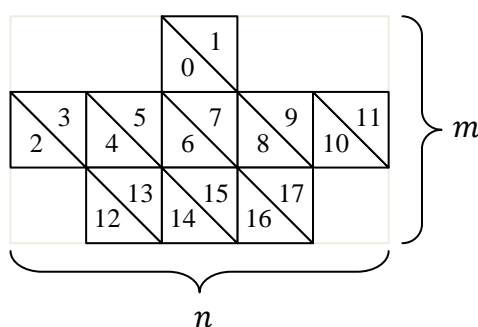
**Obr. 5.2: Jednoduchý graf oblasti  $\Omega$**

Graf je vytvořen na základě hodnot uložených v textovém souboru „kostra\_grafu.txt“. Struktura souboru je následující:

první řádek	výška oblasti $y$	šířka oblasti $x$
druhý řádek	počet řádků elementů $m$	maximální počet elementů na řádku $n$
třetí řádek	index začátku 1. řádku	index konce 1. řádku
čtvrtý řádek	index začátku 2. řádku	index konce 2. řádku

**Obr. 5.3: Struktura souboru kostra\_grafu**

Na každém řádku v souboru jsou uloženy dvě číselné hodnoty oddělené mezerou, na prvním řádku mohou být hodnoty typu *double*, na dalších řádcích pak musí být pouze hodnoty typu *int*. Šířka a výška oblasti jako taková neovlivní strukturu grafu. Hodnota  $m$  určuje, kolik dalších řádků se musí v souboru vyskytovat. Na každém dalším řádku jsou indexy, které udávají, kolik elementů na řádku bude. Zadávané indexy musí být v rozsahu od 0 do  $n - 1$ . Tato obecná struktura umožňuje vytvářet i jiné než obdélníkové pole. Pokud zadáme např.  $m = 3, n = 10$  a následně zvolíme tři řádky v těchto intervalech  $\langle 4,5 \rangle, \langle 0,9 \rangle, \langle 2,7 \rangle$ , získáme takovýto graf:



**Obr. 5.4: Složitější nepravidelný graf oblasti  $\Omega$**

Jednotlivé vrcholy grafu – elementy oblasti  $\Omega$  jsou očíslovány po řádcích zleva doprava od 0 do 17, což představuje 18 elementů. Toto očíslování je důležité pro pozdější určení sousedních prvků jednotlivých elementů, stejně jako objemů, porozity a rychlostí toku látky mezi elementy oblasti  $\Omega$ .

Na základě tohoto grafu bychom mohli uvažovat libovolně komplikovanou dvourozměrnou oblast  $\Omega$ , kde by se látka mohla transportovat zleva doprava a zároveň mezi jednotlivými řádky grafu.

### 5.3 Určení toků a objemů elementů

Pro každý element v grafu si musíme uložit informace o tom, s kterými elementy sousedí a jaká je rychlost toku látky na hranách mezi elementy. Velikost toku z elementu  $E_i$  do  $E_j$  určíme podle vztahu  $q_{ij}^+ = v \frac{x}{m}$ , kde  $v$  je makroskopická rychlost toku látky oblastí  $\Omega$ ,  $x$  je výška oblasti a  $m$  počet řádků elementů. Rychlost toku  $q$  je stejná pro všechny elementy sítě. Skutečná mikroskopická rychlost proudění látky mezi jednotlivými elementy však závisí na jejich objemech a porozitě. Proto si pro každý element musíme uložit informaci o jeho objemu. Pro zjednodušení předpokládáme, že velikost všech elementů je stejná, a proto bude i jejich objem konstantní na celé oblasti. Objem elementu  $|E_i|$  tak závisí na geometrické výšce a šířce oblasti  $\Omega$  a na celkovém počtu elementů v oblasti. Spočteme ho takto:

$$|E_i| = \frac{dx \cdot dy}{2}, \quad (5.1)$$

kde  $dx$  a  $dy$  je šířka a výška jednoho elementu. Šířku elementu spočteme jako podíl geometrické šířky oblasti a počtu elementů na řádku, výšku elementu pak jako podíl geometrické výšky oblasti a počtu řádků elementů ve sloupci oblasti  $\Omega$ .

### 5.4 Řídká transportní matice

Z vypočtených hodnot porozity, objemů a toků mezi elementy můžeme podle vztahu (3.7) vytvořit transportní matici  $A$ . Jelikož transportní matice pro trojúhelníkové pole  $n$  elementů může obsahovat maximálně tři prvky na řádku  $i$ , v ostatních sloupcích řádku  $i$  budou nuly. Proto by ukládání všech hodnot matice i následné operace s maticí byly značně neefektivní. Z těchto důvodů je matice  $A$  uložena jako řídká matice [3] a to následujícím způsobem. Pro každý řádek  $i$  matice se vytvoří dvě pole: *pole indexů* a

*pole hodnot*. V *poli indexů* jsou uloženy indexy sloupců, ve kterých se na řádku  $i$  vyskytuje nenulový prvek. Pro každý index  $j$ , uložený v *poli indexů* na pozici  $k$ , je na pozici  $k$  v *poli hodnot* uložena příslušná hodnota, vyskytující se v matici  $A$  na řádku  $i$ , ve sloupci  $j$ . Pro pochopení uvedeme, jak by vypadala tato dvě pole pro třetí řádek (index řádku  $i = 2$ ) matice z obrázku 3.2 na straně 12.

	$j \rightarrow$						
		0	1	2	3	4	$pole\ index\dot{u} = [1, 2]$
$i = 2$		0	5	-5	0	0	$pole\ hodnot = [5, -5]$

**Obr. 5.5:** Uložení jednoho řádku řídké matice

Z těchto dvou polí pro řádek  $i = 2$  lze snadno zjistit, že ve sloupci  $j = 1$  se nalézá hodnota 5 a ve sloupci  $j = 2$  hodnota -5.

Obě dvě pole musíme samozřejmě vytvořit pro každý řádek  $i$  matice  $A$ . K tomuto účelu můžeme použít dvourozměrné pole indexů a pole hodnot, kde na indexu  $i$  uložíme referenci na konkrétní pole pro řádek  $i$ .

#### 5.4.1 Rozdělení transportní matice

Pro úspěšné otestování metody AFVM potřebujeme rozdělit transportní matici  $A$  podle schématu (4.1) na straně 14. Rozdělit správně řídkou matici však není tak triviální. Nejprve musíme vytvořit permutační vektor, který pro staré indexy řádků a sloupců matice  $A$  vytvoří nové tak, abychom po přeuspořádání prvků v matici  $A$  na základě permutačního vektoru získali matice  $A_1$  a  $A_2$ .

#### 5.4.2 Tvorba permutačního vektoru a dělení matice $A$

Definujeme si dvě proměnné  $new\_1 = 0$  a  $new\_2 = n - 1$ , kde  $n$  značí počet řádků matice  $A$ , které udávají pozici v nové matici  $A_p$ . Dále nalezneme maximální prvek  $max$  na diagonále matice  $A$ . Jelikož jsou na diagonále záporné hodnoty, bude  $max$  rovno absolutní hodnotě z nejmenšího prvku na diagonále matice (pro matici  $A$  na

obrázku 5.6 je  $max = 81$ ). Permutační vektor bude představovat pole hodnot typu *int*, které uložíme jako *perm\_vektor*. Poté procházíme každý řádek *i* matice *A* a každou nenulovou hodnotu  $A_{ij}$  ve sloupci *j* porovnáme s  $max/2$ . Pokud je  $|A_{ii}| > max/2$ , pak na pozici *i* v permutačním vektoru uložíme hodnotu *new\_1*, jejíž hodnotu následně zvýšíme o jedna. Pokud je  $|A_{ii}| \leq max/2$ , pak na pozici *i* v permutačním vektoru uložíme hodnotu *new\_2*, kterou následně o jedna snížíme.

Mějme například tuto transportní matici *A* (pro dvourozměrnou oblast  $\Omega$ ):

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	-81				81								
1	81	-81											
2		81	-81										
3				-36					36				
4				36	-81	45							
5						-52	52						
6			81				-81						
7							29	-29					
8									-36	36			
9										-36	36		
10						7					-36	29	
11												-29	29
12								29					-29

Obr. 5.6: Transportní matice *A*

Výše zmíněným průchodem maticí *A* z obrázku 5.6 získáme tento permutační vektor:

indexy	0	1	2	3	4	5	6	7	8	9	10	11	12
permutace	0	1	2	12	3	4	5	11	10	9	8	7	6

Obr. 5.7: Permutační vektor pro matici *A*

Nyní projdeme opět matici *A* a každý prvek  $A_{ij}$  přidáme do nové matice  $A_p$  na řádek *perm\_vektor*[*i*] do sloupce *perm\_vektor*[*j*]. Nově vzniklá matice  $A_p$  je zobrazena na obrázku 5.8.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	-81			81									
1	81	-81											
2		81	-81										
3				-81	45								36
4					-52	52							
5			81			-81							
6							-29					29	
7							29	-29					
8					7			29	-36				
9									36	-36			
10										36	-36		
11						29						-29	
12											36		-36

**Obr. 5.8: Permutovaná matice  $A_p$**

Tučně ohraničené oblasti obsahují jen hodnoty, které jsou buď menší nebo větší než je hodnota  $max/2$ . Horní levá oblast je de facto naše hledaná matice  $A_1$  a dolní pravá oblast reprezentuje matici  $A_2$ . Po takovémto rozdělení by nám ale zůstalo pár hodnot mimo tyto dvě matice. Levou dolní část matice  $A$  proto musíme připojit k matici  $A_1$ , zatímco pravou horní část k matici  $A_2$ . Abychom k  $A_1$  nemuseli přidávat všechny řádky levé dolní oblasti, z nichž je většina prázdných (ve skutečnosti obsahují nuly), tak si ještě dolní oblast mírně upravíme. Pro prvek  $A_{8,4} = 7$  prohodíme celý řádek na indexu 8 s řádkem na indexu  $new\_1 = 6$  a hodnotu  $new\_1$  zvětšíme o jedna. Poté ještě prohodíme sloupec 6 a sloupec 8, aby záporné diagonální prvky byly na správných místech. To samé pak provedeme s prvkem  $A_{11,5} = 29$ , kdy prohodíme řádky a sloupce na indexech 11 a  $new\_1 = 7$ . Pro horní pravou oblast provedeme podobné úpravy, jen potřebné řádky prohazujeme s řádky na indexech  $new\_2$ , kdy hodnotu  $new\_2$  po každém prohození o jedna snížíme.



Po další úpravě, popsané na straně 23, vypadá matice  $A_p$  takto:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	-81					81							
1	81	-81											
2		81	-81										
3			81	-81									
4				52	-52								
5					45	-81							36
6					7		-36					29	
7				29				-29					
8								29	-29				
9							36			-36			
10										36	-36		
11									29			-29	
12											36		-36

**Obr. 5.9: Permutovaná matice  $A_p$  s prohozenými prvky**

Z matice  $A_p$  na obrázku 5.9 pak snadno vytvoříme dvě nové matice dle znázorněných čar,  $A_1$  o rozměrech  $8 \times 6$  a matici  $A_2$  s rozměry  $8 \times 7$ .

Výše popsaný postup samozřejmě nemusíme v každém kroku aplikovat na matici  $A$  a vytvářet matici novou. Po naplnění permutačního vektoru stačí projít prvky matice  $A$  a zjistit, které řádky a sloupce musíme prohodit. Poté zaměníme příslušné indexy v permutačním vektoru a takto upravený permutační vektor můžeme využít pro rozklad matice  $A$ .

Upravený permutační vektor by pro náš příklad vypadal takto:

indexy	0	1	2	3	4	5	6	7	8	9	10	11	12
permutace	<b>0</b>	<b>1</b>	<b>2</b>	<b>12</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>7</b>	<b>10</b>	<b>9</b>	<b>6</b>	<b>11</b>	<b>8</b>

**Obr. 5.10: Upravený permutační vektor**

Tento vektor dále využijeme k přiřazení původních indexů pro nové indexy matic  $A_1$  a  $A_2$ . Jelikož obě nově vytvořené matice indexujeme od nuly, musíme si pro každý řádek pamatovat, na jakém indexu se v původní matici  $A$  nacházel. To je důležité, pokud

bychom potřebovali složit zpět matici  $A$ , nebo při násobení matice krát vektor, kde nám to např. umožní násobit malou maticí  $A_1$  celý vektor koncentrací pro původní matici  $A$ .

Řádky matic  $A_1$  a  $A_2$  tak očíslováme na základě jejich umístění v matici  $A_p$ . Např. řádek na indexu  $i_p = 10$  v matici  $A_p$ , neboli šestý řádek matice  $A_2$  ( $i_2 = 5$ ), bude mít index původního řádku roven indexu  $i_{per}$  permutačního vektoru, pro který platí, že  $perm\_vektor[i_{per}] = i_p$ . Pro matici  $A_2$  vytvoříme *pole\_puv\_radku* a uložíme do něj index  $i_{per} = 8$  na pozici  $i_2 = 5$  (*pole\_puv\_radku*[5] = 8).

Takto potom vypadají rozdělené matice  $A_1$  a  $A_2$  včetně indexů původních řádků:

		$j \rightarrow$						
<i>pole_puv_radku</i> [ $i$ ]			0	1	2	3	4	5
0	$i$ ↓	0	-81					81
1		1	81	-81				
2		2		81	-81			
6		3			81	-81		
5		4				52	-52	
4		5					45	-81
10		6					7	
7		7				29		

Obr. 5.11: Matice  $A_1$

		$j \rightarrow$							
<i>pole_puv_radku</i> [ $i$ ]			0	1	2	3	4	5	6
10	$i$ ↓	0	-36					29	
7		1		-29					
12		2		29	-29				
9		3	36			-36			
8		4				36	-36		
11		5			29			-29	
3		6					36		-36
4		7							36

Obr. 5.12: Matice  $A_2$

Kdybychom pak potřebovali najít např. původní pozici prvku  $A_{1,6,4} = 7$ , využijeme *pole\_puv\_radku* v matici  $A_1$  na obrázku 5.11. Původní řádek prvku je roven *pole\_puv\_radku*[6] = 10, původní sloupec se rovná *pole\_puv\_radku*[4] = 5.

### 5.4.3 Násobení řídké matice vektorem

Permutovanou řídkou matici nyní reprezentují tyto tři pole: *pole\_indexu*[], *pole\_hodnot*[], a nově také *pole\_puv\_radku*{}. Třetí pole s indexy původních řádků nám umožňuje ukládat matice s řídkou řádkovou strukturou

Uvedeme si rovnou kód metody *matice\_X\_vektor*, napsaný v jazyce Java.

```

1. public static double [] matice_X_vektor(SparseMatrix A,double [] vektor){
2.     double [] vysledny_vektor = new double [vektor.length];
3.     double vysledek;
4.     for (int i=0; i<A.pole_indexu.length;i++){
5.         vysledek = 0;
6.         for(int j=0;j<A.pole_indexu[i].length;j++){
7.             int index_sloupce = A. pole_indexu[i][j];
8.             if (index_sloupce != -1){
9.                 vysledek += A.pole_hodnot[i][j] *
10.                    vektor[A.getPuvodniRadek()][index_sloupce]];
11.             }
12.         }
13.         vysledny_vektor[A.getPuvodniRadek()][i]] = vysledek;
14.     }
15.     return vysledny_vektor;
16. }
```

**Obr. 5.13: Kód metody *matice\_X\_vektor***

Parametry metody jsou reference na řídkou matici  $A$  a na *vektor*, což je pole hodnot typu *double*. Metoda vrací *vysledny\_vektor*, jehož velikost je stejná jako velikost vektoru *vektor*. Metoda prochází řídkou matici po řádcích, kde pro každý řádek  $i$  projde všechny *indexy* sloupců uložené v *pole\_indexu*[ $i$ ]. Pokud je *index* v *pole\_indexu*[ $i$ ][ $j$ ]  $\geq 0$  (což znamená, že prvek matice  $A_{i,index} \neq 0$ ), pak je jeho hodnota, uložená na příslušné pozici v *pole\_hodnot*[ $i$ ][ $j$ ], vynásobena hodnotou

vektoru na pozici *pole\_puv\_radku[index]*, která odpovídá původnímu sloupci prvku  $A_{i,index}$ . Tato hodnota se přičte do proměnné *vysledek*. Po průchodu všech prvků na řádku *i* se hodnota proměnné *vysledek* zapíše do vektoru *vysledny\_vektor* na pozici *pole\_puv\_radku[i]* a následně se *vysledek* vynuluje.

Tato metoda tak umožňuje násobit např. malou matici  $A_1$  vektorem koncentrací *c*, jehož velikost odpovídá počtu řádků původní transportní matice *A*. Hodnoty v matici  $A_1$  budou vynásobeny jen těmi řádky vektoru *c*, které odpovídají původním řádkům matice  $A_1$ . Výsledkem bude řádkový vektor o velikosti vektoru *c*, obsahující vypočtené hodnoty pouze na těch indexech, které odpovídají indexům původních řádků matice  $A_1$ . Na ostatních řádcích vektoru *c* budou nuly.

## 6 Testování implementovaných metod

Tato kapitola se věnuje testování implementovaných metod FVM a AFVM. První dva testy mají ověřit, jestli se látka v prostředí dostane za určitý čas na určené místo. Třetí test kontroluje celkovou hmotnost transportované látky v čase, zda zůstává zachován zákon o zachování hmoty. Čtvrtý test porovnává obě implementované metody a poslední test je zaměřen na srovnání výpočetní času obou metod.

### 6.1 Testovací úloha

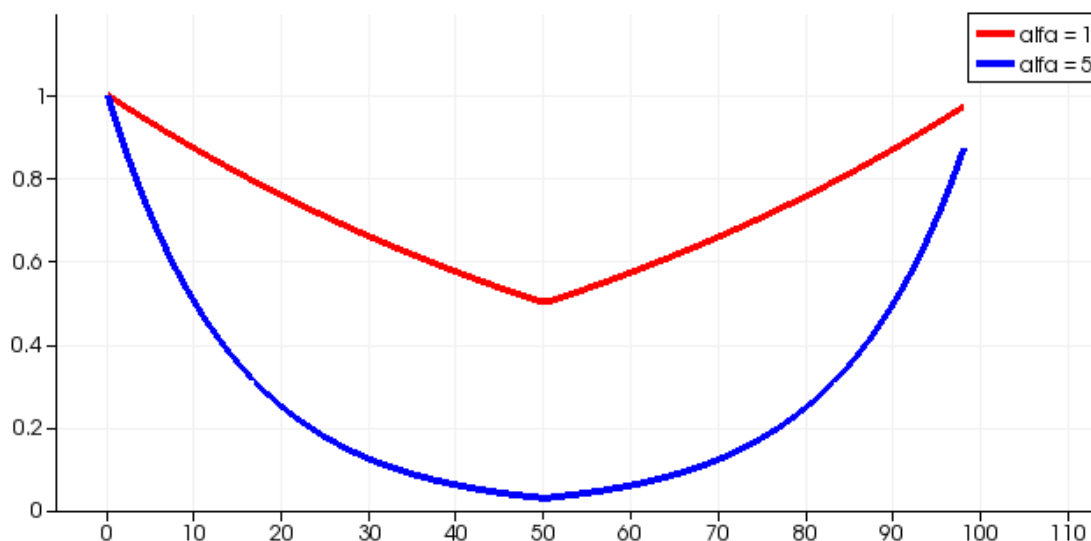
Pro snadnější ověření správné funkčnosti a porovnání obou metod budeme uvažovat pouze jednorozměrnou oblast  $\Omega$ , kde látka proudí zleva doprava, k transportu mezi řádky grafu nedochází. Oblast  $\Omega$  má rozměry  $0,5 \times 1$ , makroskopická rychlost proudění látky  $v = 5$ . Ve většině testů uvažujeme diskretizaci oblasti tak, že počet elementů na řádku grafu je  $n = 1000$ , počet řádků ve sloupci je  $m = 2$ . Zároveň zavedeme periodické okrajové podmínky, které nám zaručí, že jakmile se látka dostane do posledního elementu na řádku  $i$ , transportuje se v dalším kroku metody do prvního elementu na řádku  $i$ . Tím de facto z řádku  $i$  vytvoříme uzavřenou oblast, kde bude látka proudit kolem dokola. Zde použité dva řádky elementů slouží jen jako dva identické proudy, které využijeme kvůli lepší vizualizaci v programu Paraview.

Jako poslední potřebujeme určit hodnoty porozity jednotlivých elementů. Jelikož naše nová metoda počítá s rozdělenou transportní maticí, kde  $A_1$  má být malá matice obsahující pouze vysoké toky, vytvoříme pro výpočet porozity takovou funkci, která by nám na začátku oblasti generovala vysoké hodnoty a uprostřed oblasti naopak řádově nižší hodnoty. Nízké hodnoty porozity uprostřed oblasti  $\Omega$  nám pak zaručí řádově vyšší rychlosti toku látky na příslušných elementech. K tomuto výpočtu použijeme exponenciální funkci o základu 2, v jejímž exponentu bude figurovat koeficient  $\alpha$ . Porozitu  $\gamma_i$  na elementu  $i$  vypočteme podle rovnic (6.1) a (6.2).

$$\gamma_i = 2^{\alpha \left(1 - 2\left(\frac{i}{n}\right)\right) - \alpha} \quad \text{pro } i \leq \frac{n}{2}, \quad (6.1)$$

$$\gamma_i = 2^{\alpha(2(\frac{i}{n})-1)-\alpha} \quad \text{pro } i > \frac{n}{2}, \quad (6.2)$$

kde  $n$  je počet elementů a  $i$  je konkrétní element  $E_i$ . Na obrázku 6.1 jsou ukázány dvě křivky spojitě funkce pro výpočet porozity pro  $n = 100$ , v prvním případě je  $\alpha = 1$ , ve druhém pak  $\alpha = 5$ .



**Obr. 6.1: Graf porozity pro  $\alpha = 1$  a  $\alpha = 5$**

Hodnoty na ose  $x$  přísluší indexům  $i$  jednotlivých elementů, jejichž rozsah je  $\langle 0,99 \rangle$ .

## 6.2 Test doby oběhu pro $\alpha = 0$

Koeficient  $\alpha$  je důležitou součástí rovnice pro výpočet porozity. Když je  $\alpha = 0$ , spočítaná porozita  $\gamma_i = 1$  pro každý element  $i$  oblasti  $\Omega$ . Jelikož pro naši úlohu jsou objemy všech elementů stejné, bude všude stejná i mikroskopická rychlost toku látky. Vektor koncentrací  $c_i = 1$  pro elementy v intervalu  $I_c = \langle 0; 0,05 \rangle$  na ose  $x$ .

Předpokládáme, že za těchto podmínek by se měla koncentrace látky šířit konstantní rychlostí, aniž by se vektor  $c$  rozmazal. Daná koncentrace by měla dorazit zpět do počátečního intervalu za čas  $t = 0,2$ .

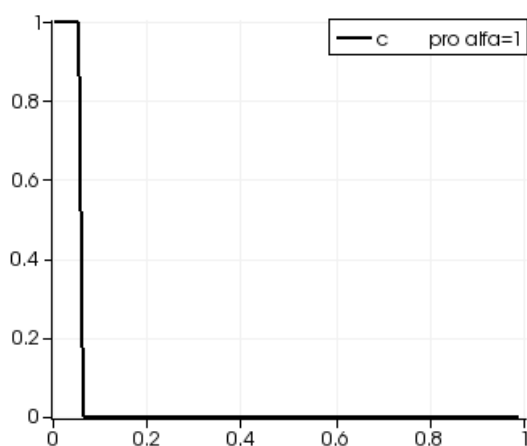
Po provedení příslušných výpočtů jsme ověřili, že doba oběhu byla skutečně  $t = 0,2$  a hodnoty koncentrací se neměnily, pouze se posouvaly ve směru toku látky.

Výstupy metody FVM a AFVM byly naprosto shodné, křivka koncentrací v čase  $t = 0,2$  vypadala jako křivka na obrázku 6.2.

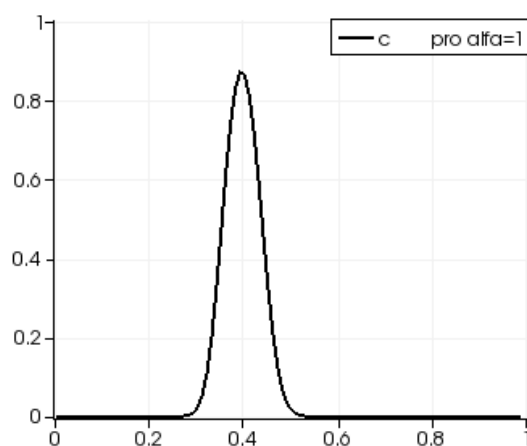
### 6.3 Test doby oběhu pro $\alpha \geq 0$

Vektor koncentrací nastavíme stejně jako v předchozím testu, hodnotu  $\alpha$  budeme měnit. Grafy uvedené v tomto testu přísluší hodnotám vypočteným metodou FVM. Test byl úspěšně proveden i pro metodu AFVM.

Nejprve otestujeme, kam doběhne koncentrace látky za čas  $t = 0,2$ , pro  $\alpha = 1$ .



Obr. 6.2: Počáteční stav koncentrace



Obr. 6.3: Stav koncentrace v čase  $t=0,2$

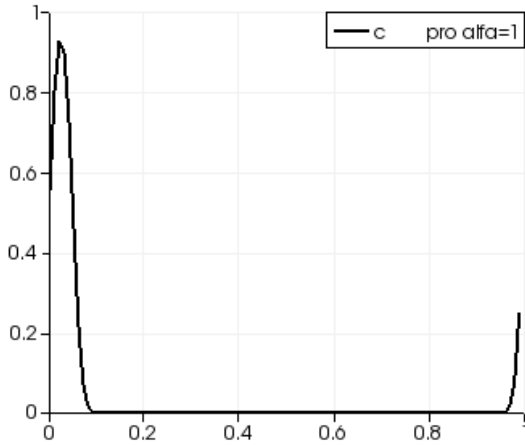
Pro  $t = 0,2$  a  $\alpha = 1$  jsme zjistili, že koncentrace látky oběhla jednou kolem dokola a při druhém oběhu se zastavila kousek před polovinou oblasti  $\Omega$ , jak je vidět na obrázku 6.3. Z toho plyne, že čas oběhu je ovlivněn velikostí koeficientu  $\alpha$ .

Řešením diferenciální rovnice  $\frac{dx}{dt} = \frac{v}{\gamma(x)}$  lze určit čas jednoho oběhu pro  $\alpha \neq 0$  dle vztahu (6.3)

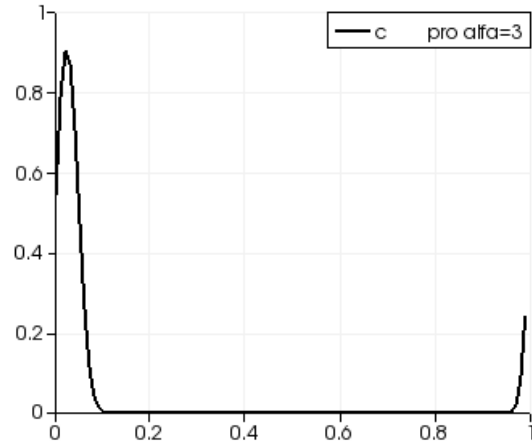
$$T = 2(1 - 2^{-\alpha}) \frac{1}{\ln 4\alpha v} l_x, \quad (6.3)$$

kde  $l_x$  je délka oblasti ve směru osy  $x$ . Pro naši úlohu, kde  $\alpha = 1$ , vypočteme  $T = 2(1 - 2^{-1}) \frac{1}{\ln 4 \cdot 5} 1 \cong 0,14427$ . Zkusíme tedy otestovat, zda koncentrace za tento

čas dorazí do původní polohy. Zároveň otestujeme stejnou úlohu pro  $\alpha = 3$ , pro kterou je čas jednoho oběhu  $T \cong 0,084157$ .



Obr. 6.4:  $c$  po jednom oběhu,  $\alpha = 1$



Obr. 6.5:  $c$  po jednom oběhu,  $\alpha = 3$

Z obou dvou grafů je vidět, že největší hodnota koncentrace se dostala na začátek oblasti  $\Omega$ . Po podrobnější analýze jednotlivých hodnot obou grafů jsme zjistili, že nejvyšší hodnota koncentrace pro  $\alpha = 1$ ,  $c = 0,926438$  se nachází na pozici  $x = 0,02$ , což je přibližně uprostřed našeho počátečního intervalu koncentrací  $I_c$ . Nejvyšší hodnota  $c = 0,904493$  pro  $\alpha = 3$  se vyskytovala na stejném místě. Vypočtený čas jednoho oběhu koncentrací byl tedy správný pro obě hodnoty  $\alpha$ , takže náš test byl úspěšný.

## 6.4 Test balance hmoty

Tento test je zaměřen na to, jestli se celková hmotnost látky proudící oblastí  $\Omega$  nějak výrazně nemění v čase. Jelikož je naše oblast uzavřená, látka by se neměla nikde ztrácet, ani vznikat.

Hmotnost látky elementu na indexu  $i$  vypočteme podle následujícího vzorce:

$$m_{E_i} = c_i \cdot |E_i| \cdot \gamma_i \quad (6.4)$$



kde  $c_i$  je koncentrace látky na elementu  $E_i$ ,  $|E_i|$  je velikost objemu  $E_i$  a  $\gamma_i$  je jeho porozita. Pro celou oblast  $\Omega$  by mělo platit, že  $\sum_{i=0}^n m_{E_i}$  bude stejná jak v čase  $t$ , tak v čase  $t + T$ , kde  $T$  je doba jednoho oběhu koncentrace látky uzavřenou oblastí  $\Omega$ .

Proto porovnáme celkovou hmotnost látky na počátku simulace ( $m_t$ ) a na jejím konci ( $m_{t+T}$ ), tj. po jednom oběhu. Z těchto dvou hodnot vypočteme relativní chybu  $e$ :

$$e = \frac{m_t - m_{t+T}}{m_t} \quad (6.5)$$

V tomto testu porovnáme hodnoty  $e$  pro obě implementované metody.

$\alpha$	$e_{FVM}$	$e_{AFVM}$
0	0	0
1	$1,127 \cdot 10^{-15}$	$1,408 \cdot 10^{-16}$
3	$-1.3561 \cdot 10^{-15}$	$4.5204 \cdot 10^{-16}$
5	$3,1226 \cdot 10^{-14}$	$-4,8288 \cdot 10^{-16}$
8	$1.8123 \cdot 10^{-13}$	$-1.7716 \cdot 10^{-16}$

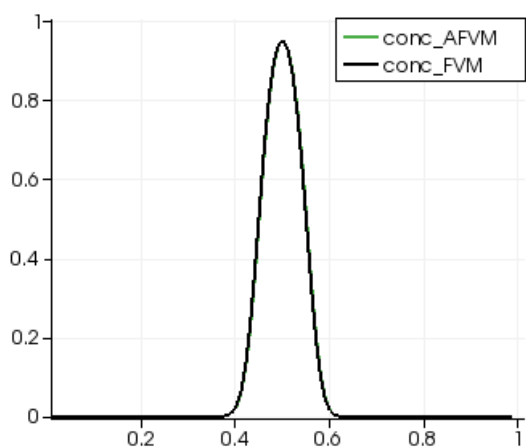
**Obr. 6.6: Tabulka relativních chyb pro různá  $\alpha$**

Z tabulky na obrázku 6.6 lze vypožorovat, že relativní chyby jsou velmi malé, takže můžeme říct, že bilance hmoty je zachována. Zajímavé však je, že relativní chyba původní metody FVM se se zvyšujícím se  $\alpha$  zvyšuje, kdežto chyba pro naši novou metodu AFVM zůstává řádově stejná.

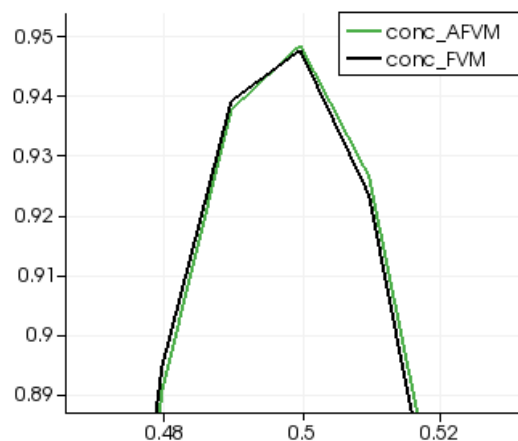
## 6.5 Porovnání implementovaných metod

Jak už bylo vidět na grafech v předchozích testech, dochází při výpočtu koncentrací k „rozmazávání“. Interval na ose  $x$ , určující v kolika elementech je  $c_i > 0$ , se rozšiřuje, zatímco maximální hodnota koncentrace v tomto intervalu se snižuje. Tento jev je způsoben tzv. numerickou difúzí, jejíž míra závisí na metodě pro výpočet koncentrací.

Pro lepší porovnání numerické difuze nastavíme počáteční koncentrace  $c_i = 1$  pro elementy v intervalu  $I_c = \langle 0,45; 0,55 \rangle$ . Ostatní parametry zůstanou stejné, koeficient  $\alpha = 1$ , ve druhém příkladu je pak  $\alpha = 5$ .



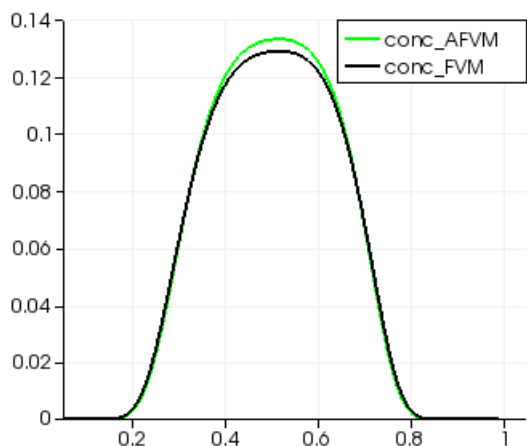
Obr. 6.7: Porovnání koncentrací po jednom oběhu



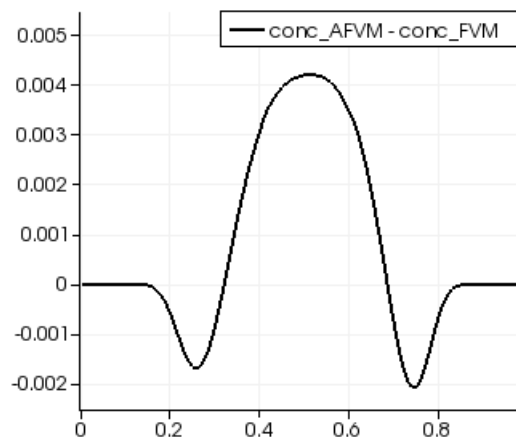
Obr. 6.8: Zvětšená "špička" grafu na obrázku 6.7

Na těchto dvou obrázcích je vidět, že křivky koncentrací vypočtené oběma metodami vypadají totožně. Při zvětšení už nepatrný rozdíl vidět je. Porovnáním zobrazených křivek zjistíme, že maximální hodnota koncentrace je vyšší pro AFVM.

K mnohem větší numerické difúzi však dochází při větším  $\alpha$ . Např. pro  $\alpha = 5$ :



**Obr. 6.9: Viditelně odlišné koncentrace**



**Obr. 6.10: Rozdíl koncentrací vypočtených oběma metodami**

Rozdíl v hodnotách koncentrací (obrázek 6.10) je patrný i na obrázku 6.9. Z grafů vyplývá, že nová metoda AFVM méně „rozmazává“, tj. uprostřed oblasti  $\Omega$ , kde  $c_i > 0$ , jsou hodnoty koncentrací vyšší než u FVM, kdežto na okrajích jsou hodnoty nižší, jak je vidět z grafu na obrázku 6.10.

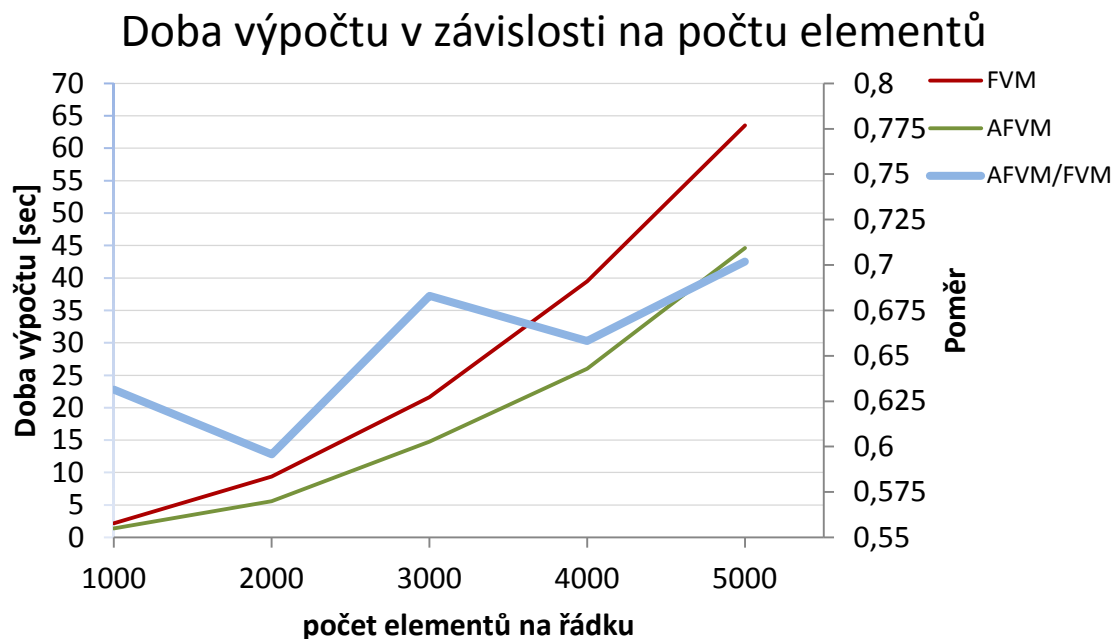
Porovnáním obou metod na základě výše uvedených grafů můžeme konstatovat, že metoda AFVM funguje správně, poskytuje podobné výsledky jako původní metoda FVM a navíc s menší mírou numerické difúze.

## 6.6 Porovnání výpočetních časů

Jako poslední musíme otestovat, jak dlouho trvá počítači výpočet koncentrací původní metodou FVM a naší novou metodou AFVM. Jelikož nová metoda pracuje s rozdělenou transportní maticí, kdy s maticí  $A_1$  počítá dvakrát tak často než s maticí  $A_2$ , měl by výpočet metodou AFVM trvat kratší dobu. Obě metody používají stejné maticové násobení.

Naměřené hodnoty jsem získal tak, že před začátkem a na konci výpočtu si program uložil aktuální hodnoty času v nanosekundách, následně je od sebe odečetl a výsledný čas vypsál do konzole. Tento čas je proto spíše orientační, jeho hodnota závisí na způsobu implementace obou testovaných metod a na použitém hardware.

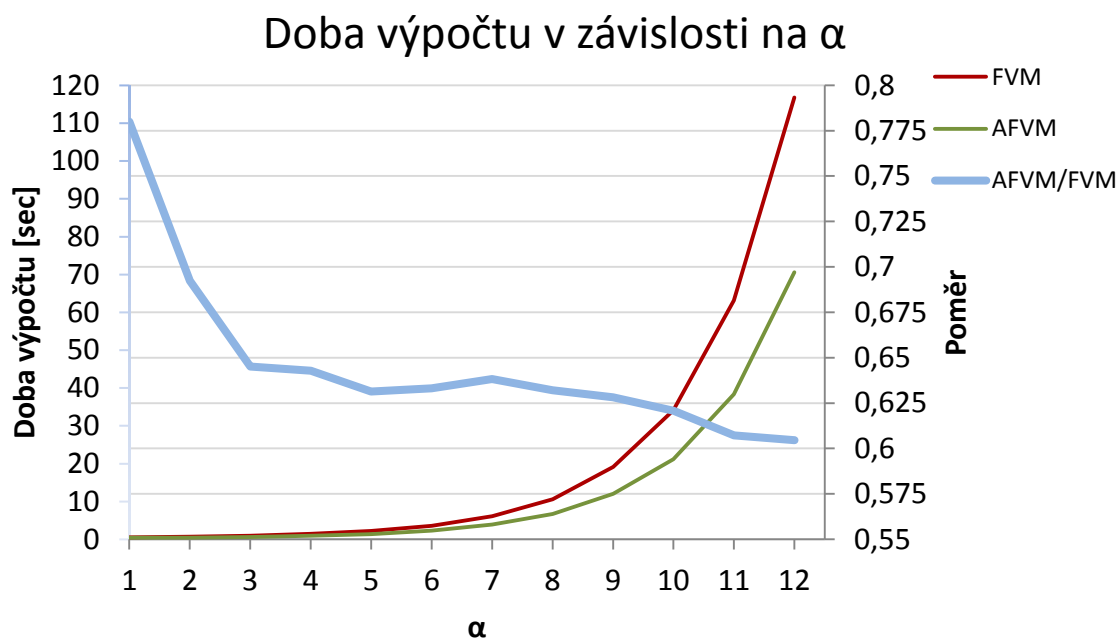
Nejprve otestujeme výpočetní čas pro různě veliké pole elementů na řádku oblasti  $\Omega$ . Budeme tedy měnit parametr  $n$ , kdy koncentrace spočítáme pro hodnoty  $n = 1000$ ,  $n = 2000$  atd. až po  $n = 5000$ . Konstanta  $\alpha = 5$ . Zjištěné doby výpočtu obou metod odpovídají deseti oběhům počátečního vektoru koncentrací.



**Obr. 6.11:** Graf výpočetních časů pro různou diskretizaci oblasti

Z grafu je vidět, že výpočet metodou AFVM trvá skutečně kratší dobu, a že křivka AFVM roste pomaleji než křivka FVM. Poměr výpočetních časů kolísá přibližně mezi hodnotami 0,6 a 0,7, k viditelnému zlepšení nedochází.

Nyní otestujeme, jak závisí doba výpočtu na parametru  $\alpha$ . Test byl proveden na testovací úloze kde  $n = 1000$  a  $\alpha$  nabývá celočíselných hodnot v intervalu  $\langle 1,12 \rangle$ . Připomeňme si, že parametr  $\alpha$  určuje hodnotu porozity  $\gamma$  na elementech. Čím nižší je porozita na daném elementu, tím rychlejší je skutečná rychlost proudění látky v tomto místě. Rozdílný rozsah rychlostí pak ovlivní velikosti transportních matic  $A_1$  a  $A_2$ , což by se mělo projevit při výpočtu koncentrací metodou AFVM.



**Obr. 6.12:** Graf výpočetních časů pro různé hodnoty  $\alpha$

Obě křivky mají exponenciální průběh, pro hodnotu  $\alpha = 12$  je doba výpočtu metodou AFVM téměř dvakrát kratší než u FVM. Se zvyšujícím se  $\alpha$  dochází ke zmenšení transportní matice  $A_1$ , takže výpočet koncentrací metodou AFVM pak trvá mnohem kratší dobu.

## 7 Závěr

Po provedení všech popsaných testů můžeme konstatovat, že nová metoda pro výpočet koncentrací konečných objemů AFVM funguje správně pro jednorozměrnou uzavřenou oblast  $\Omega$ . K jednomu oběhu transportované látky dochází za správný čas, celková hmotnost látky se v průběhu simulace nemění. Nová metoda také splňuje uvedené předpoklady – poskytuje hodnoty koncentrací s nižší numerickou difúzí a výpočet těchto hodnot trvá kratší dobu než výpočet původní metodou FVM. Pro dosažení ještě lepších výsledků metodou AFVM bychom mohli dále dělit matici  $A_2$  stejným způsobem, jakým jsme rozdělili původní matici  $A$ .

Otázkou ovšem zůstává, jak bude metoda AFVM fungovat na dvourozměrných, či trojrozměrných oblastech. Zejména by se mělo ověřit, zda matice, aproximující časový dvojkrok, splňuje CFL podmínku pro řádné rychlostní pole. K tomuto zkoumání by se dal využít algoritmus pro tvorbu jednoduchého dvourozměrného grafu s volitelnou šířkou a výškou, který jsem implementoval do svého programu.

Dále by bylo vhodné implementovat novou metodu AFVM do programu Flow123d, který nabízí mnohem širší možnosti simulace proudění a transportu látek v podzemním prostředí. Po implementaci bychom tak mohli provést řadu dalších testů, které by správnost nové metody konečných objemů ověřily.

## Seznam použité literatury

- [1] CÍSLEROVÁ Milena, VOGEL Tomáš. *Transportní procesy*. 1. vydání, Vydavatelství ČVUT, 1998. ISBN 80-01-01866-0
  
- [2] MARYŠKA Jiří, HOKR Milan, KRÁLOVCOVÁ Jiřina a ŠEMBERA Jan. *Modelování transportních procesů v horninovém prostředí*. 1. vydání, Technická univerzita v Liberci, 2010. ISBN 978-80-7372-571-6
  
- [3] *Sparse Matrix Storage Formats* [online], [cit. 17. května 2012], URL: [http://software.intel.com/sites/products/documentation/hpc/compilerpro/en-us/cpp/win/mkl/refman/appendices/mkl\\_appA\\_SMSF.html](http://software.intel.com/sites/products/documentation/hpc/compilerpro/en-us/cpp/win/mkl/refman/appendices/mkl_appA_SMSF.html)